

Prácticas

Sistemas Inteligentes I

Sesión 8. Redes Neuronales
José A. Montenegro Montes
monte@lcc.uma.es

Resumen

- Ejemplo `aima.gui.demo.learning.LearningDemo.java`
- Práctica Redes Neuronales



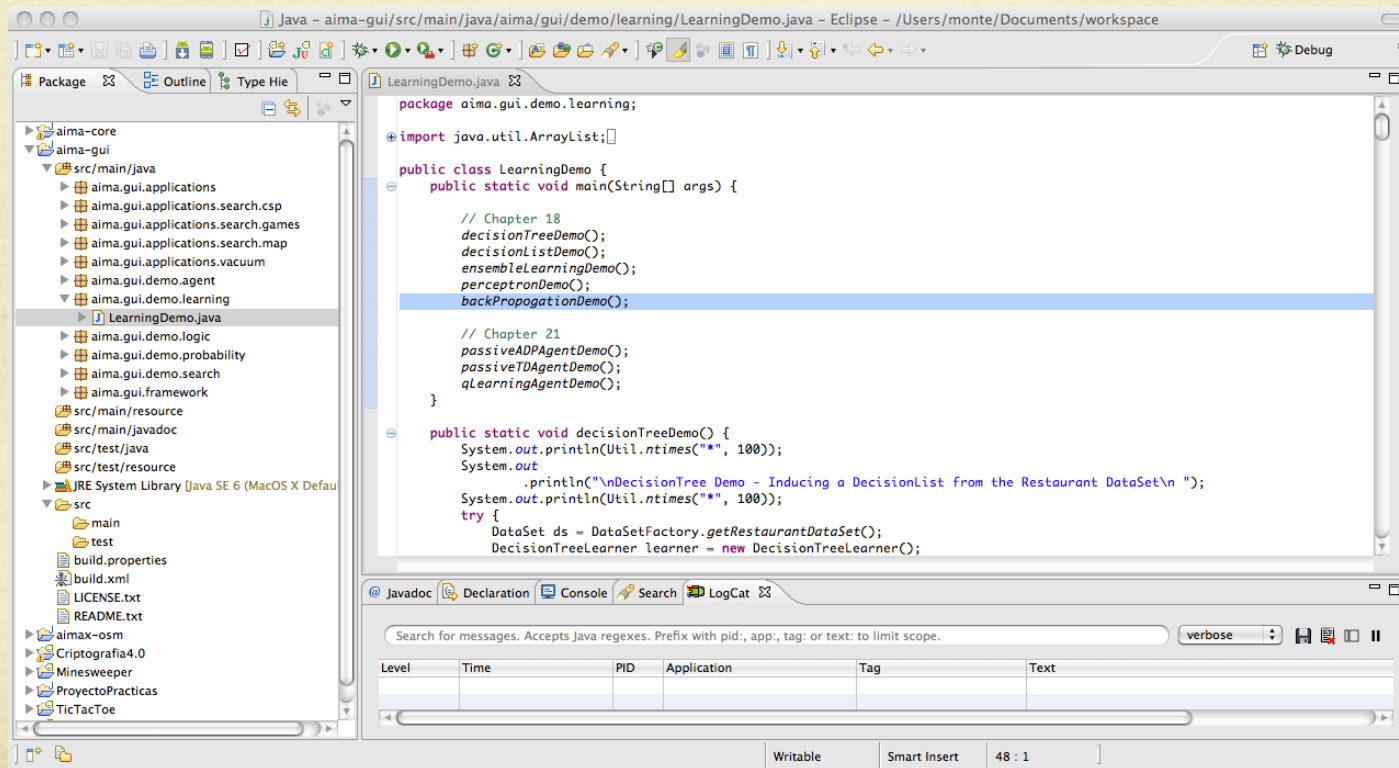
AIMA

Ejemplo

aima.gui.demo.learning.LearningDemo.java

aima.gui.demo.learning.LearningDemo.java

El fichero *aima.gui.demo.learning.LearningDemo.java* hay un ejemplo de como configurar una red multicapa en el método *backPropogationDemo*.



```
package aima.gui.demo.learning;

import java.util.ArrayList;

public class LearningDemo {
    public static void main(String[] args) {

        // Chapter 18
        decisionTreeDemo();
        decisionListDemo();
        ensembleLearningDemo();
        perceptronDemo();
        backPropogationDemo();

        // Chapter 21
        passiveADPAgentDemo();
        passiveTDAgentDemo();
        qLearningAgentDemo();
    }

    public static void decisionTreeDemo() {
        System.out.println(Util.ntimes("**", 100));
        System.out
            .println("\nDecisionTree Demo - Inducing a DecisionList from the Restaurant DataSet\n ");
        System.out.println(Util.ntimes("**", 100));
        try {
            DataSet ds = DataSetFactory.getRestaurantDataSet();
            DecisionTreeLearner learner = new DecisionTreeLearner();
```


aima.gui.demo.learning.LearningDemo.java

```
public static void backPropogationDemo() {  
  
    .....  
    NNConfig config = new NNConfig();  
    config.setConfig(FeedForwardNeuralNetwork.NUMBER_OF_INPUTS, 4);  
    config.setConfig(FeedForwardNeuralNetwork.NUMBER_OF_OUTPUTS, 3);  
    config.setConfig(FeedForwardNeuralNetwork.NUMBER_OF_HIDDEN_NEURONS,6);  
    config.setConfig(FeedForwardNeuralNetwork.LOWER_LIMIT_WEIGHTS, -2.0);  
    config.setConfig(FeedForwardNeuralNetwork.UPPER_LIMIT_WEIGHTS, 2.0);  
  
    FeedForwardNeuralNetwork ffnn = new FeedForwardNeuralNetwork(config);  
    ffnn.setTrainingScheme(new BackPropLearning(0.1, 0.9));  
    ffnn.trainOn(innds, 10);  
  
    .....  
}
```



AIMA

Práctica

Redes Neuronales

Práctica Redes Neuronales

Configurar una red neuronal multicapa (*feedforward*) y ver su comportamiento para aproximar funciones (squared cosine y humps).

Para ello, se deberá de terminar el método *prepare* de la clase *NeuralController* (fichero *NeuralApp.java*).

- ¿Cuántas épocas son necesarias para aprender cada una de las funciones?
- ¿Qué tipo de red (10, 20 o 50 neuronas ocultas) aprende más rápido?

Práctica Redes Neuronales

```
public void prepare(String changedSelector) {
    epochs = 0;
    if(changedSelector != null) clear();
    else {
        try {
            NNDataSet ds = new FunctionNNDataSet(
                frame.getSelection().getValue(NeuralFrame.DATASET_SEL));

            // Set the neural architecture NNConfig
            // NUMBER_OF_INPUTS,NUMBER_OF_OUTPUTS,LOWER_LIMIT_WEIGHTS,UPPER_LIMIT_WEIGHTS
            switch(frame.getSelection().getValue(NeuralFrame.NEURAL_SEL)) {
                case 0: // 10 hidden neurons
                    break;
                case 1: // 20 hidden neurons
                    break;
                case 2: // 50 hidden neurons
                    break;
            }
        }
    }
}

cont...
```

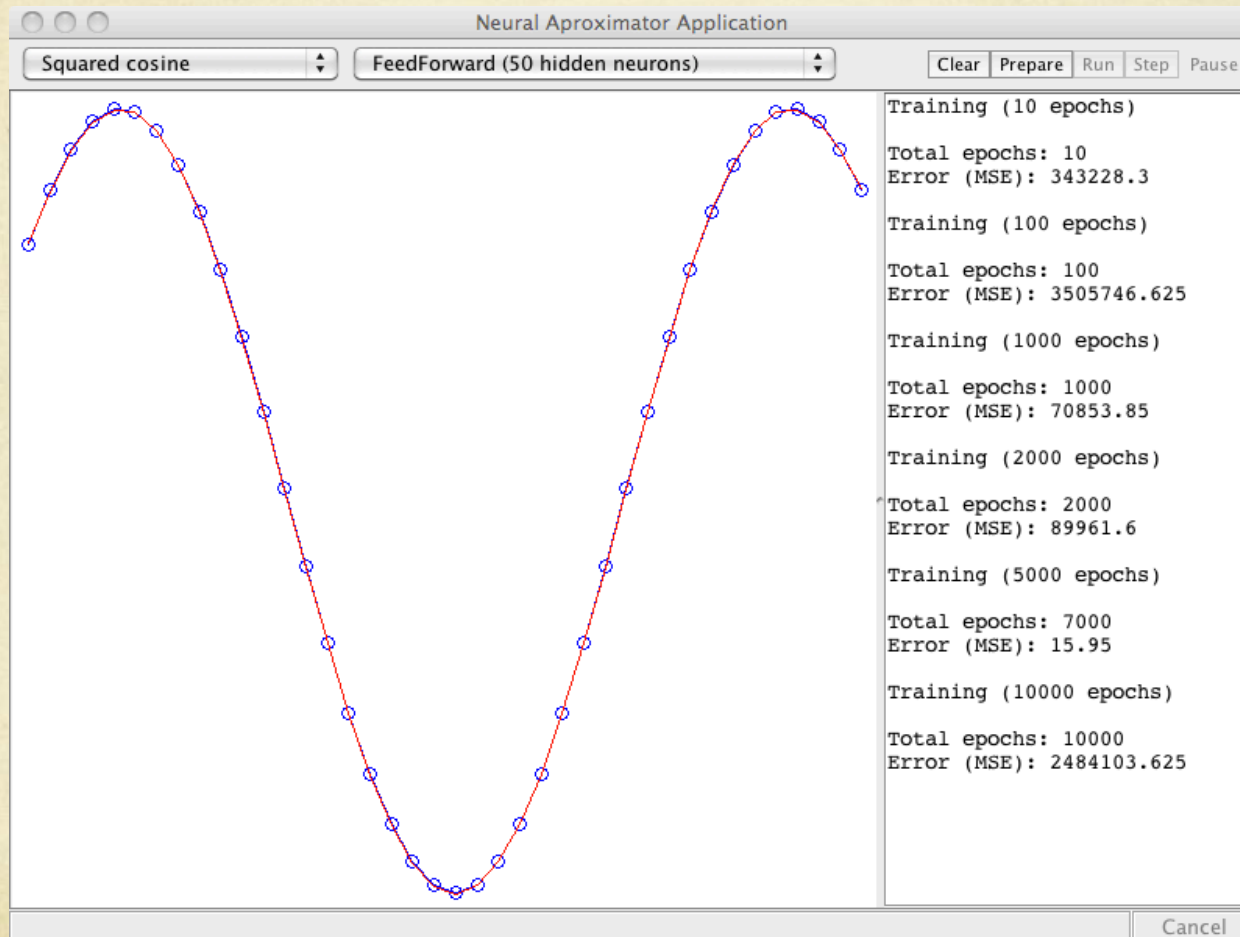

Práctica Redes Neuronales

```
// Create the neural network and training scheme
```

```
// FeedForwardNeuralNetwork ffnn
```

```
        env = new NeuralEnvironment(ds, ffnn);  
        frame.getEnvView().setEnvironment(env);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}  
}
```

Práctica Redes Neuronales



Práctica Redes Neuronales

